LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-SR-524004

Report on subcontract from Lawrence Livermore National Lab, "Development of Large-Dimension Configuration-Interaction Shell-Model Code"

C. W. Johnson

January 24, 2012

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Report on subcontract from Lawrence Livermore National Lab, "Development of Large-Dimension Configuration-Interaction Shell-Model Code"

PI: Professor Calvin Johnson, SDSU

(LLNL technical contact: Dr. W. Erich Ormand)

The project period was devoted to several developments in the technical capabilities of the BIGSTICK large-dimension configuration-interaction shell-model code, written in Fortran 90.

The specific computational goals for the project period were:
(1) Store Lanczos vectors on core in RAM to minimize I/O.
(2) Rewrite reorthogonalization with Lanczos vectors stored in core; consult with personnel at LLNL, LBL, ORNL, Iowa State University to maximize performance.
(3) Restrict creation of N-body jumps to those needed by an individual node;
(4) Distribute 3-body interaction over many cores.

We made significant progress towards these goals, especially (1) and (2), although in the process we discovered intermediate tasks that had to be accomplished first. Our achievements were as follows:
-- I put into place structures and algorithms to facility fragmenting very large-dimension Lanczos intermediate vectors. Only by fragmenting the vectors can we carry out (1) and (2).
-- In addition, I reorganied the action of the Hamiltonian matrix and created a new division of operations for MPI. Based upon earlier work, I made plans (in consultation with Dr. Ormand) of a revised algorithm for distribution of work with MPI, with a particular eye towards breaking up the Lanczos vectors. I introduce a new derived type ("opbundles") which collects the parameters for the Hamiltonian, and rewrote the application routines to use it. It has been validated and verified.
--I made orogress towards revised MPI parallelization. Using the "opbundles," I was able to compute a distribution of work over compute nodes, which should be very efficient. This new distribution is easier to derive and more efficient, in principle, than our old distribution. Furthermore, it should make applications with fragmented Lanczos vectors easier. Implementation is still in progress.

Additional achievements, not listed in the statement of work, but which proved necessary to move towards our goals:
-- Revised application of OpenMP parallel coding, including 3-body. Using subroutines written by Dr. Ormand, I made some minor simplifications to our use of OpenMP, including to 3-body interactions, which leads to significant speed-up.
-- Conversion of Lanczos vectors from double precision to single precision. Discussions with Dr. Pieter Maris of Iowa State University convinced us the Lanczos vectors could be represented with single-precision rather than double precision, even for very large dimensions, as long as some intermediate operations were carried out in double precision. (Dr. Ormand had anticipated this possibility and we wrote the code with this flexibility in mind.) I modified the code so that it could be easily switched between single- and double-precision and confirmed, at least for medium sized problems (dimensions around 100 million) good agreement with no undesireable loss of precision. Further tests may be desired for 1-10 billion, but this is now easy.
This change will help further work as for large dimensions it (a) speeds up write/read of Lanczos vectors to disk (b) reduces RAM storage of Lanczos vectors and (c) will speed up internal communication of Lanczos vectors when using MPI.
-- Implementation of option for "time-reverse" storage of uncoupled two-body matrix elements. Hamiltonian matrix elements are read in coupled form, then internally decoupled via Clebsch-Gordan

coefficients. There is a simple symmetry between matrix elements with all m-values switched in sign, with an overall phase. Using this storage option reduces the internal storage requirements for the decoupled matrix elements, at a minor cost by using an additional phase. The real savings will be in using the same strategy for storing decoupled three-body matrix elements (currently in progress, almost finished).

Although I did not fully achieve the stated goals, I made significant progress towards them and have significantly improved the performance of the current code.